Remarks and Arguments

Claims 1-36 were presented for examination. Claims 1, 7, 9, 13, 15, 16, 18, 24, 26, 30, 32, 33, 35 and 36 have been amended. Claims 12 and 29 have been canceled.

Claims 1-7, 11-13, 17-24, 28-30 and 34 were rejected under 35 U.S.C. §102(e) as anticipated by U.S. Patent No. 6,397,268 (Cepulis.) The examiner comments that the Cepulis references discloses the recited limitations.

The Cepulis reference discloses a mechanism for tracking PCI bus numbers that change during a configuration. As is well-known, the PCI bus system discussed in Cepulis is a memory mapped system. In particular, each PCI device in the system, including each PCI-PCI bus bridge, has a predetermined set of addresses in a PCI address space. The location of the address set for a particular PCI device depends on the PCI bus to which that device is connected and the PCI socket on that bus. Although the PCI addresses are in the address space of the computer, reads and writes to these addresses actually read and write registers in the PCI device associated with that address set. This is described in Cepulis at column 4, lines 17-34. Consequently, a PCI device will only respond when commands are sent to its PCI address.

Thus, in order to "walk" a PCI bus, as described in Cepulis at column 11, lines 21-37, it is only necessary to check for a PCI device at each of the PCI address sets in the PCI address space. Generally, this is done by attempting to read one of the registers, for example, the register that holds the vendor ID. If the vendor ID can be read, then a device exists at that address. If an error occurs then no device is located at that address. Once a device is discovered at an address, the configuration software can determine whether the device is a PCI-PCI bus bridge by reading the class code register which has a predetermined value for PCI-PCI bus bridge devices. Once a PCI-PCI bus bridge is discovered, its address is known and further configuration commands are sent directly to that address. After the discovered PCI-PCI bus bridge has been configured, the configuration software then continues walking the busses by attempting reads to addresses in the PCI address space. For a more complete discussion of PCI bus configuration, the examiner is referred to www.tldp.org/LDP/tlk/dd/pci.html.

The present invention is directed to an I^2C or "wired-AND" bus. This type of bus is not memory mapped and, thus, the mechanism for "walking" the bus disclosed in

Cepulis cannot be used in a wired_AND bus system because there is no address space from which device addresses can be determined. Consequently, a completely different system is used to "walk" the busses and assign bus bridge IDs. In particular, each bridge is designed to respond only to configuration commands sent to its assigned bridge ID. The bridge ID thus acts as like the PCI address in Cepulis. The configuration software initially sets the bridge IDs of all bridges to a predetermined bridge ID value. Then, the configuration software sends a configuration command to this predetermined bridge ID value. A bridge on the highest level bus will respond to this configuration command since its bridge ID value has been initialized to the predetermined value. Note that only one bridge on the highest level bus will respond to the configuration command because of the CGF OUT/CFG IN connections between the bridges at each bus level. Once a bus has been discovered, in this manner, its bridge ID value is set to another bridge ID value different from the predetermined value. Then further configuration commands can be sent to this new bridge ID value in order to configure that bridge and any other bridges connected to that bridge. After a bridge has been configured, the CFG OUT/CFG IN connections cause another bridge on the highest level to respond to the predetermined bridge ID value. In this manner, by repeatedly sending configuration commands to the predetermined bridge ID, the entire bus network can be configured.

The claims have been amended to particularly point out this operation. For example, claim 1 has been amended to recite that each bridge responds to configuration commands sent to its bridge ID and in step (a), "initially setting the bridge ID of all bridges to a predetermined bridge ID value and walking the bus system to discover the bus topology and the bus bridges that form that topology by repeatedly sending commands and data to the predetermined bridge ID value" and, in step (b), "assigning a unique bridge ID different from the predetermined bridge ID value to each discovered bridge. As previously mentioned, the PCI system disclosed in Cepulis does not set the addresses of all PCI-PCI bus bridges to the same predetermined value. Instead, the addresses of the bus bridges are determined by the bus they are on and their position on that bus. Similarly, Cepulis does not "walk" a bus by repeatedly sending commands to the same bridge ID value. Instead Cepulis "walks" a bus by

sequentially testing addresses in the PCI address space. Once a bus bridge is discovered, configuration commands are sent to the bridge address located in the PCI address space.

Accordingly, Cepulis discloses a mechanism for walking the PCI busses that differs substantially from the claimed mechanism. It is noted that the examiner asserts that Cepulis disclosed the limitations recited in original claim 12 and 29 related to mechanisms for walking a bus by setting the bridge ID of all bridges to a predetermined value and then successively configuring each bridge by sending commands and data to the predetermined bridge ID at column 8, lines 45-67. Applicant's review of this section of Cepulis indicates that it states that a PCI/SCSI bus adapter, which is used to transfer the operating system from disk into the computer during startup, is configured by commands stored in a PCI BIOS section of the ROM BIOS. However, at column 8, line 65-column 9, line 4, Cepulis discloses that the PCI BIOS is controlled by configuration values stored in NVRAM 142 by a system configuration program. It is a system configuration program of this type with which the present invention is concerned. Cepulis does not state that all PCI-PCI bus bridges are assigned the same address, since the address is determined by the location of the bridge. Nor does Cepulis disclose configuring each bridge by sending commands and data to the predetermined bridge ID because, in Cepulis, each bridge address is identified by another means and then commands and data are sent to that address, not some common predetermined address. The examiner appears to be confusing the primary, secondary and subordinate bus values used to configure a PCI bus bridge with the bridge ID of the present invention. However, primary, secondary and subordinate bus values determine how a PCI bus bridge will respond to a configuration command that it receives not whether the PCI bridge will receive the command at all.

Consequently, amended claim 1 patentably distinguishes over the Cepulis reference. Claims 18, 35 and 36 have been amended in a similar manner to claim 1 and thus, distinguish over the Cepulis reference in the same manner. Claims 7, 9, 13, 15, 16, 24, 26, 30, 32 and 33 have been amended to conform them to the "value" term used in amended claims 1 and 18, which term is believed to more particularly point out the operation of the invention.

Claims 2-6 and 11 and 19-23 and 28 are dependent on amended claims 1 and 18, respectively, and distinguish over the Cepulis reference in the same manner as amended claim 1 and 18.

The examiner asserts that claims 7 and 24, which recite assigning a bridge ID to an upstream bridge, are disclosed in Cepulis at column 11, lines 22-37. However, this section of Cepulis recites that the configuration software updates subordinate bus numbers in upstream bridges. As previously discussed, subordinate bus numbers are not the equivalent of bridge IDs because they do not determine whether the bridge responds to configuration commands. Consequently, claims 7 and 24 patentably distinguish over the Cepulis reference.

The examiner also asserts the Cepulis, at column 4, lines 5-7 and 17-34, discloses connecting all bridges on the same hierarchical level so that only one bridge at a time responds to the predetermined bridge ID value as recited in claim 13 and 30. As previously mentioned, in Cepulis, each bridge is sent configuration commands at its own unique address. Thus, there is no need for the bridges to be connected so that only one bridge responds because no two bridges have the same address and there are no configuration commands to which two bridges could simultaneously respond. At column 4, lines 5-7 and 17-34, Cepulis discloses that there is at least one PCI bus connected to the host bus via a Host/PCI bridge and that this bus is called bus number zero and that each PCI device connected to bus zero is addressable at PCI bus number zero and that PCI devices have registers for holding information concerning the devices. This section of Cepulis is consistent with the fact stated above that the register addresses are dependent on the bus to which at device is connected and the socket on that bus. It does not indicate that more than one bridge could respond to the same address as recited in claims 13 and 30.

Regarding clams 17 and 34, the deterministic arbitration protocol recited therein is relevant only to wired-AND busses to prevent a collision problem between competing master devices and has no equivalent in PCI busses. It relates to forwarding master-specific addresses at the start of an I^2C transaction. The section of Cepulis (column 4, lines 11-61) asserted by the examiner to disclose this operation instead discusses numbering of PCI busses and how this numbering changes when an additional PCI-PCI

bus bridge is connected to the system. Thus, clam 17 and 34 patentably distinguish over the cited reference.

Claims 14-15 and 31-32 have been rejected as obvious under 35 U.S.C. §103(a) over Cepulis in view of U.S. Patent No. 6,260,092 (Story.) The examiner asserts that story discloses all of the claimed limitations except the limitation that bridges on the same level are connected in a daisy chain configuration. However, the examiner claims that story discloses bridges connected in a daisy chain. The examiner concludes that it would have been obvious to combine Cepulis and Story to reduce interconnect signal line count.

First, as discussed above, there is no need to modify Cepulis to prevent more than one bridge from responding to a command issued to an address because, in Cepulis, each command is issued to a unique address and only the bridge at that address will respond. Thus, one skilled in the art would not be motivated to seek a modification as disclosed in Story. In addition, while Story discloses ring-connectable bus bridges, it is not directed to bus configuration. Instead, Story is addressed to methods and apparatus for connecting a PCI bus system to a high speed serial link, such as a Fibre Channel. Therefore, Story itself would not suggest a combination with Cepulis. Further, even if Cepulis and Story were combined as suggested by the examiner, the combination would not teach or suggest the limitations recited in claims 14-15 and 31-32. For example, claim 14, which is dependent on claim 13, recites that all bridges on the same hierarchical level are connected in a daisy chain so that only one bridge at a time responds to the predetermined bridge ID value. In Story, the bus bridges are connected in a daisy chain, but not so that only one bridge responds to an address value. Rather the bridges are connected in a daisy chain that passes signals from one bridge to another until the bridge to which the signals are addressed responds. Thus, the combination of Story with Cepulis cannot teach or suggest this limitation because neither reference discloses or suggest it. Claim 15 further recites that the bridge ID must be changes in one bridge before another bridge can respond to the predetermined bridge ID. Story does not disclose this type of operation and neither does Cepulis. Thus claims 14 and 15 patentably distinguish over the cited references.

Claims 31 and 32, which contain similar limitations also distinguish over the cited references in the same manner.

Claims 16 and 33 have been rejected as obvious under 35 U.S.C. 103(a) over Cepulis in view of U.S. Patent No. 6,205,147 (Mayo.) The examiner indicates that Cepulis does not disclose a configuration in which two unidirectional bridges are connected in parallel, but Mayo discloses such a combination.

The PCI bus bridges disclosed in Cepulis are clearly bi-directional, thus one skilled in the art would not be motivated to seek additional references that disclose two unidirectional bridges connected in parallel. Further, Mayo is directed to packet-switched networks rather than the parallel PCI busses disclosed in Cepulis. Thus, one skilled in the art would certainly not look to Mayo for combination with Cepulis. Finally, both claims 16 and 33 recite that the two unidirectional bus bridges are assigned different bridge IDs. Neither Cepulis, nor Mayo discloses this limitation and, consequently, neither can the combination of Cepulis with Mayo teach or suggest this limitation. Thus, claims 16 and 33 patentably distinguish over the cited references.

In light of the forgoing amendments and remarks, this application is now believed in condition for allowance and a notice of allowance is earnestly solicited. If the examiner has any further questions regarding this amendment, he is invited to call applicants' attorney at the number listed below. The examiner is hereby authorized to charge any fees or direct any payment under 37 C.F.R. 1.17, 1.16 to Deposit Account number 02-3038.

Respectfully submitted

Paul E. Kudirka Date: 3/10/04

Paul E. Kudirka, Esq. Reg. No. 26,931
KUDIRKA & JOBSE, LLP
Customer Number 021127
Tel: (617) 367-4600 Fax: (617) 367-4656